P⊕RTAL
USPTO

Search: ⊙ The ACM Digital Library   ○ The Guide

| +multiprocessor, +allocation, +sequential +memory, +bypass | **SEARCH** |

⌨ Feedback   Report a problem   Satisfaction survey

Terms used
multiprocessor allocation sequential memory bypass operating system

Found 217 of 171,143

| Sort results by | relevance ▽ |
| Display results | expanded form ▽ |

❖ Save results to a Binder
? Search Tips
☐ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200   **Result page: 1   2   3   4   5   6   7   8   9   10   next**
Best 200 shown   Relevance scale ☐◻◼◼◼

---

1   Experience Using Multiprocessor Systems—A Status Report   ◼
Anita K. Jones, Peter Schwarz
June 1980 **ACM Computing Surveys (CSUR)**, Volume 12 Issue 2
Publisher: ACM Press
Full text available: 📄 pdf(4.48 MB)   Additional Information: full citation, references, citings, index terms

---

2   Performance evaluation of memory consistency models for shared-memory   ◼
multiprocessors
Kourosh Gharachorloo, Anoop Gupta, John Hennessy
April 1991 **ACM SIGARCH Computer Architecture News , ACM SIGOPS Operating**
Systems Review , ACM SIGPLAN Notices , Proceedings of the fourth
international conference on Architectural support for programming
languages and operating systems ASPLOS-IV, Volume 19 , 25 , 26 Issue 2 , Special
Issue , 4
Publisher: ACM Press
Full text available: 📄 pdf(1.71 MB)   Additional Information: full citation, references, citings, index terms

---

3   Characterizing the caching and synchronization performance of a multiprocessor   ◼
operating system
Josep Torrellas, Anoop Gupta, John Hennessy
September 1992 **ACM SIGPLAN Notices , Proceedings of the fifth international**
conference on Architectural support for programming languages and
operating systems ASPLOS-V, Volume 27 Issue 9
Publisher: ACM Press
Full text available: 📄 pdf(1.52 MB)   Additional Information: full citation, references, citings, index terms

---

4   Cache coherence in large-scale shared-memory multiprocessors: issues and   ◼
comparisons
David J. Lilja
September 1993 **ACM Computing Surveys (CSUR),** Volume 25 Issue 3
Publisher: ACM Press

Full text available: pdf(3.12 MB)   Additional Information: <u>full citation</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

5 Architectural primitives for a scalable shared memory multiprocessor

Joonwon Lee, Umakishore Ramachandran
June 1991 **Proceedings of the third annual ACM symposium on Parallel algorithms
and architectures**
Publisher: ACM Press
Full text available: pdf(1.27 MB)   Additional Information: <u>full citation</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

6 Cache Memories

Alan Jay Smith
September 1982 **ACM Computing Surveys (CSUR)**, Volume 14 Issue 3
Publisher: ACM Press
Full text available: pdf(4.61 MB)   Additional Information: <u>full citation</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

7 Supporting dynamic data structures on distributed-memory machines

Anne Rogers, Martin C. Carlisle, John H. Reppy, Laurie J. Hendren
March 1995 **ACM Transactions on Programming Languages and Systems (TOPLAS)**,
Volume 17 Issue 2
Publisher: ACM Press

Full text available: pdf(2.05 MB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index
terms</u>, <u>review</u>

Compiling for distributed-memory machines has been a very active research area in recent
years. Much of this work has concentrated on programs that use arrays as their primary
data structures. To date, little work has been done to address the problem of supporting
programs that use pointer-based dynamic data structures. The techniques developed for
supporting SPMD execution of array-based programs rely on the fact that arrays are
statically defined and directly addressable. Recursive data s ...

Keywords: dynamic data structures

8 Hiding memory latency using dynamic scheduling in shared-memory multiprocessors

Kourosh Gharachorloo, Anoop Gupta, John Hennessy
April 1992 **ACM SIGARCH Computer Architecture News , Proceedings of the 19th
annual international symposium on Computer architecture ISCA '92**, Volume
20 Issue 2
Publisher: ACM Press

Full text available: pdf(1.65 MB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index
terms</u>

The large latency of memory accesses is a major impediment to achieving high
performance in large scale shared-memory multi-processsors. Relaxing the memory
consistency model is an attractive technique for hiding this latency by allowing the overlap
of memory accesses with other computation and memory accesses. Previous studies on
relaxed models have shown that the latency of write accesses can be hidden by buffering
writes and allowing reads to bypass pending writes. Hiding the latency of ...

9
HFS: a performance-oriented flexible file system based on building-block
compositions

Orran Krieger, Michael Stumm
August 1997 **ACM Transactions on Computer Systems (TOCS)**, Volume 15 Issue 3
Publisher: ACM Press

Full text available: pdf(383.87 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>, <u>review</u>

The Hurricane File System (HFS) is designed for (potentially large-scale) shared-memory multiprocessors. Its architecture is based on the principle that, in order to maximize performance for applications with diverse requirements, a file system must support a wide variety of file structures, file system policies, and I/O interfaces. Files in HFS are implemented using simple building blocks composed in potentially complex ways. This approach yields great flexibility, allowing an application ...

Keywords: customization, data partitioning, data replication, flexibility, parallel computing, parallel file system

10 The interaction of software prefetching with ILP processors in shared-memory systems

Parthasarathy Ranganathan, Vijay S. Pai, Hazim Abdel-Shafi, Sarita V. Adve
May 1997 **ACM SIGARCH Computer Architecture News , Proceedings of the 24th annual international symposium on Computer architecture ISCA '97**, Volume 25 Issue 2
Publisher: ACM Press

Full text available: pdf(2.44 MB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

Current microprocessors aggressively exploit instruction-level parallelism (ILP) through techniques such as multiple issue, dynamic scheduling, and non-blocking reads. Recent work has shown that memory latency remains a significant performance bottleneck for shared-memory multiprocessor systems built of such processors.This paper provides the first study of the effectiveness of software-controlled non-binding prefetching in shared memory multiprocessors built of state-of-the-art ILP-based proces ...

11 External memory algorithms and data structures: dealing with **massive data**

Jeffrey Scott Vitter
June 2001 **ACM Computing Surveys (CSUR)**, Volume 33 Issue 2
Publisher: ACM Press

Full text available: pdf(828.46 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

Data sets in large applications are often too massive to fit completely inside the computers internal memory. The resulting input/output communication (or I/O) between fast internal memory and slower external memory (such as disks) can be a major performance bottleneck. In this article we survey the state of the art in the design and analysis of external memory (or EM) algorithms and data structures, where the goal is to exploit locality in order to reduce the I/O costs. We consider a varie ...

Keywords: B-tree, I/O, batched, block, disk, dynamic, extendible hashing, external memory, hierarchical memory, multidimensional access methods, multilevel memory, online, out-of-core, secondary storage, sorting

12 Scalable lock-free dynamic memory allocation

Maged M. Michael
June 2004 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2004 conference on Programming language design and implementation PLDI '04**, Volume 39 Issue 6

Publisher: ACM Press

Full text available: pdf(213.94 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

Dynamic memory allocators (malloc/free) rely on mutual exclusion locks for protecting the consistency of their shared data structures under multithreading. The use of locking has many disadvantages with respect to performance, availability, robustness, and programming flexibility. A lock-free memory allocator guarantees progress regardless of whether some threads are delayed or even killed and regardless of scheduling policies. This paper presents a completely lock-free memory allocator. It uses ...

Keywords: async-signal-safe, availability, lock-free, malloc

13 Trace-driven memory simulation: a survey

Richard A. Uhlig, Trevor N. Mudge
June 1997 **ACM Computing Surveys (CSUR)**, Volume 29 Issue 2
Publisher: ACM Press

Full text available: pdf(636.11 KB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>, <u>review</u>

As the gap between processor and memory speeds continues to widen, methods for evaluating memory system designs before they are implemented in hardware are becoming increasingly important. One such method, trace-driven memory simulation, has been the subject of intense interest among researchers and has, as a result, enjoyed rapid development and substantial improvements during the past decade. This article surveys and analyzes these developments by establishing criteria for evaluating trac ...

Keywords: TLBs, caches, memory management, memory simulation, trace-driven simulation

14 GTW: a time warp system for shared memory multiprocessors

Samir Das, Richard Fujimoto, Kiran Panesar, Don Allison, Maria Hybinette
December 1994 **Proceedings of the 26th conference on Winter simulation**
Publisher: Society for Computer Simulation International
Full text available: pdf(860.15 KB)   Additional Information: <u>full citation</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

15 Instruction prefetching of systems codes with layout optimized for reduced cache misses

Chun Xia, Josep Torrellas
May 1996 **ACM SIGARCH Computer Architecture News , Proceedings of the 23rd annual international symposium on Computer architecture ISCA '96**, Volume 24 Issue 2
Publisher: ACM Press

Full text available: pdf(1.65 MB)   Additional Information: <u>full citation</u>, <u>abstract</u>, <u>references</u>, <u>citings</u>, <u>index terms</u>

High-performing on-chip instruction caches are crucial to keep fast processors busy. Unfortunately, while on-chip caches are usually successful at intercepting instruction fetches in loop-intensive engineering codes, they are less able to do so in large systems codes. To improve the performance of the latter codes, the compiler can be used to lay out the code in memory for reduced cache conflicts. Interestingly, such an operation leaves the code in a state that can be exploited by a new type of ...

16 Data prefetch mechanisms

Steven P. Vanderwiel, David J. Lilja
June 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 2
Publisher: ACM Press

Full text available: 📄 pdf(172.07 KB)   Additional Information: full citation, abstract, references, citings, index terms, review

The expanding gap between microprocessor and DRAM performance has necessitated the use of increasingly aggressive techniques designed to reduce or hide the latency of main memory access. Although large cache hierarchies have proven to be effective in reducing this latency for the most frequently used data, it is still not uncommon for many programs to spend more than half their run times stalled on memory requests. Data prefetching has been proposed as a technique for hiding the access lat ...

Keywords: memory latency, prefetching

17 Multigrain shared memory

Donald Yeung, John Kubiatowicz, Anant Agarwal
May 2000 **ACM Transactions on Computer Systems (TOCS)**, Volume 18 Issue 2
Publisher: ACM Press

Full text available: 📄 pdf(369.18 KB)   Additional Information: full citation, abstract, references, index terms, review

Parallel workstations, each comprising tens of processors based on shared memory, promise cost-effective scalable multiprocessing. This article explores the coupling of such small- to medium-scale shared-memory multiprocessors through software over a local area network to synthesize larger shared-memory systems. We call these systems Distributed Shared-memory MultiProcessors (DSMPs). This article introduces the design of a shared-memory system that uses multiple granularities of sharing, ca ...

Keywords: distributed memory, symmetric multiprocessors, system of systems

18 A performance study of memory consistency models

Richard N. Zucker, Jean-Loup Baer
April 1992 **ACM SIGARCH Computer Architecture News , Proceedings of the 19th annual international symposium on Computer architecture ISCA '92**, Volume 20 Issue 2
Publisher: ACM Press

Full text available: 📄 pdf(1.17 MB)   Additional Information: full citation, abstract, references, citings, index terms

Recent advances in technology are such that the speed of processors is increasing faster than memory latency is decreasing. Therefore the relative cost of a cache miss is becoming more important. However, the full cost of a cache miss need not be paid every time in a multiprocessor. The frequency with which the processor must stall on a cache miss can be reduced by using a relaxed model of memory consistency. In this paper, we present the results of instruction-level simulation s ...

19 Program transformation and runtime support for threaded MPI execution on shared-memory machines

Hong Tang, Kai Shen, Tao Yang
July 2000 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 22 Issue 4
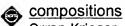**Publisher:** ACM Press

Full text available: 📄 pdf(352.21 KB)   Additional Information: full citation, abstract, references, citings, index

terms

Parallel programs written in MPI have been widely used for developing high-performance applications on various platforms. Because of a restriction of the MPI computation model, conventional MPI implementations on shared-memory machines map each MPI node to an OS process, which can suffer serious performance degradation in the presence of multiprogramming. This paper studies compile-time and runtime techniques for enhancing performance portability of MPI code running on multiprogrammed share ...

Keywords: MPI, lock-free synchronization, multiprogrammed environments, program transformation, shared-memory machines, threaded execution

20 HFS: a performance-oriented flexible file system based on building-block compositions
Orran Krieger, Michael Stumm
May 1996 **Proceedings of the fourth workshop on I/O in parallel and distributed systems: part of the federated computing research conference**
Publisher: ACM Press
Full text available: pdf(1.83 MB)   Additional Information: full citation, references, citings, index terms

Results 1 - 20 of 200   Result page: **1** 2 3 4 5 6 7 8 9 10   next

Useful downloads: Adobe Acrobat   QuickTime   Windows Media Player   Real Player

IEEE *Xplore* ℗
RELEASE 2.1

Welcome United States Patent and Trademark Office

[☐] Search Session History

BROWSE   SEARCH   IEEE XPLORE GUIDE

Fri, 17 Feb 2006, 8:29:13 PM EST

Edit an existing query or compose a new query in the Search Query Display.

Search Query Display

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Recent Search Queries

#1	((multiprocessor)<in>metadata)
#2	(allocating process<IN>metadata)
#3	(allocation<IN>metadata)
#4	(allocating and (segment or block or portion or section)<IN>metadata)
#5	((sequential or consecutive) and memory<IN>metadata)
#6	(operating system<IN>metadata)
#7	((bypass or without) and (operating system)<IN>metadata)
#8	(((multiprocessor)<in>metadata)) <AND> ((allocating process<IN>metadata))
#9	(((bypass or without) and (operating system)<IN>metadata)) <AND> ((((multiprocessor)<in>metadata)) <AND> ((allocating process<IN>metadata)))

Help   Contact Us   Privac

© Copyright 2006 IE

Indexed by
Inspec*